基于文本分析的故障序列模式挖掘算法 *

常文兵，苑星龙，周晟瀚†，李　磊

(北京航空航天大学 可靠性与系统工程学院，北京 100191)

摘　要：针对结构化程度差、表达形式各异的文本数据，提出了一种基于文本信息的故障序列模式挖掘算法，用以发掘故障之间的时序关系。为从文本记录的故障信息中挖掘故障规律，首先将文本信息向量化，对故障文本信息进行相似度衡量，将表达相同意义的故障归为一类。在此基础上根据故障特性，提出最大窗口阈值、最小共现度阈值的概念，构建故障序列模式挖掘算法框架。最后对某型飞机文本故障信息进行序列模式挖掘，找出了正确的故障序列关系。实例验证了所提算法是正确有效的。

关键词：序列模型；数据挖掘；文本相似度；飞机故障；文本挖掘

中图分类号：V267+.3　　doi: 10.3969/j.issn.1001-3695.2018.02.0142

Failures sequence pattern mining algorithm based on text analysis

Chang Wenbing, Yuan Xinglong, Zhou Shenghan†, Li Lei

(*School of Reliability & System Engineering*, *Beihang University*, *Beijing* 100191, *China*)

Abstract: For textual data with poor structured degree and different expression forms, a failures sequence pattern mining algorithm based on text information is proposed to explore the time sequence relationship between failures. In order to mine the failures rules from the text, firstly, quantify the text information, measure the similarity of the failures information, and classify the failures that express the same meaning into one class. On this basis, we propose the concept of maximum window threshold and minimum concurrence threshold based on failures characteristics, and build a mining algorithm framework forfailures sequence pattern. Finally, extract sequential failures patterns from a certain aircraft, and find out the correct faliures sequence relationship. The example shows that the proposed algorithm is correct and effective.

Key words: sequence pattern; data mining; text similarity; aircraft failure; text mining

0　引言

飞机作为大型的复杂装备系统，服役周期长，飞行环境复杂恶劣，导致飞机的故障发生频繁，原因复杂。在长期的维修保障过程中积累了大量的故障文本信息，对故障分析和维修决策具有重大意义，有待进行更深层次的挖掘。目前，还没有针对文本信息应用序列模式挖掘识别故障之间的时间关系的研究。

本文第一部分是故障文本相似度衡量，由于文本记录的特性，相同信息的表达形式千差万别，通过文本相似度衡量把相同意思的文本划归为一类。首先要对故障本文进行预处理，使用语言模型进行自然语言处理是建立在词的基础上的，由于中文与英文的区别需要先对故障文本进行分词，才能做进一步的处理。吴熠潇[1]介绍了什么是中文分词，中文分词的国内研究现状和当前的研究热点，介绍了统计语言模型,以及如何利用简化的语言模型进行中文分词。本文将采用 jieba 分词算法来进行分词处理。在分完词的基础上，要把文本信息用数学语言表达出来才能进行下一步的相似度衡量。郑文超等人[2]提出了一种中文分词算法，用来将中文文本分割成独立的词语。再对处理后的语料使用 word2vec 工具集，应用深度神经网络算法，转换为对应的词向量。张志昌等人[3]提出一种中文词汇蕴涵关系识别方法，利用词向量技术,设计各种基于词向量的分类特征,训练得到可用于名词词汇蕴涵关系分类的支持向量机分类模型.周练[4]研究了 word2vec 模型的原理及应用，分析了词向量的特点。唐明等人[5]提出了一种基于 word2Vec 模型的文档向量表示，利用 TF-IDF 算法计算每篇文档中词的权重，并结合 word2vec 词向量生成文档向量。现有的研究大都集中在词汇特征和句法特征的提取上，而忽略了词语之间的语义关系，Zhang 等人[6]为了获得语义特征，提出了一种基于 word2vec 和 SVM perf 的情感分类方法。殷耀明[7]提出了基于关系向量模型的句子相似度计算，考虑句子结构和语义信息，更能体现句子的结构和语义

收稿日期：2018-02-11；**修回日期：**2018-04-10　**基金项目：**国家自然科学基金资助项目（71501007）；航空科学基金资助项目；北航研究生教育发展基金资助项目

作者简介：常文兵，男，北京人，副研究员，博士，主要研究方向为数据挖掘；苑星龙，男，山东济宁人，硕士，主要研究方向为数据挖掘；周晟瀚（通信作者），男，北京人，讲师，博士，主要研究方向为数据挖掘（zhoush@buaa.edu.cn）；李磊，男，北京人，硕士，主要研究方向为数据挖掘.

信息。为了解决机器翻译依赖问题，同时仍然利用资源丰富的语言中的数据，Tian 等人[8]提出联合学习低资源语义的语义文本相似性任务和资源丰富语义的语义文本相似性任务，该任务仅依赖于多语言词语嵌入。本文第二部分是构建序列模式挖掘算法，也是本文核心部分。序列模式挖掘是数据挖掘的一个重要领域，苗雪连[9]描述了间隙约束序列挖掘的分类及研究现状，给出了间隙约束的序列模式挖掘在实际生活中的发展趋势并认为在未来的研究领域中,具有间隙约束的序列模式挖掘仍是一个重要的研究方向。

基于模式增长的序列模式挖掘（FreeSpan）的初步研究，Krishna[10]提出了一种更有效的方法，称为 PSP，用于高效挖掘顺序模式。Le 等人[11]研究了频繁闭合和发生器序列的开采任务，因为与频繁序列集合相比，频繁闭合和发生器序列的基数通常远低于频繁序列的基数。针对数据集的增多，约束频繁模式树的构建存在一定的缺陷，约束频繁模式树很难应用于海量数据集，Yan 等人[12]使用 MapReduce 编程模型提出了一种被称为 PACFP 的约束频繁模式的并行挖掘算法。武优西等人[13]采用模式匹配技术，在一遍扫描序列数据库的情况下，建立其所有超模式的不完整网树森林，并对这些超模式的支持率进行有效地计算，进而挖掘出所有频繁模式，有效地提高了序列模式挖掘速度。Mooney[14]专注于数据挖掘的子领域序列模式挖掘，研究迄今为止提出的方法和算法。Aloysius [15]提出了一种使用 PrefixSpan 算法挖掘用户购买模式的方法，并根据采购模式的顺序将产品放置在货架上。Wright[16]使用顺序模式挖掘来自动推断药物之间的时间关系，可视化这些关系，并生成规则以预测可能为患者开具的下一个药物。

由于本文挖掘数据是文本信息，故障序列模式的挖掘算法较传统序列模式挖掘算法发生了很大变化。首先是对文本数据的结构化处理，对不同描述的同一类故障归类，找出相似项目集；之后根据研究对象的特点在故障序列模式挖掘过程中定义最大事件窗口阈值和最小共现度阈值，在此基础上构建了故障序列模式挖掘的算法框架。

1　研究方法

针对故障文本进行挖掘的序列模式为重复有间隙型序列模式，其中间隙是指两个事件之间不是必须紧挨。因为不一定连续两次故障之间才存在因果关系，所以考虑一定间隔内按次序发生的故障序列更符合实际。以图 1 所示序列模式为例，假设最大间隙为 2，则对于序列A → B，用元素的位置表示来代表序列关系，则 S_1 中，1 → 3，2 → 3，6 → 7，6 → 8满足条件，记录 4 次，S_2 中1 → 2满足条件，记录 1 次，两条序列中共记录 5 次。如果两事件满足一定事件间隔要求，则两事件序列模式的支持度加一。

S_1 = A A B C D A B B
position 1 2 3 4 5 6 7 8

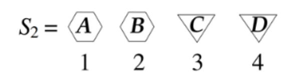


图 1 重复有间隙型序列模式示例

1.1　相关概念与定义

下面定义了概念与符号，便于对后面步骤进行准确描述：

在故障序列模式挖掘过程中定义最小相似度阈值、最小频繁度阈值、最大事件窗口阈值、最小支持度阈值和最小共现度阈值。其中最小相似度阈值指文本信息划归为同一类的最低要求，划归成的类称为相似事件集；最小频繁度阈值指相似事件集中最小事件个数，对于低于最小频繁度阈值的事件集，认为这类故障极少发生，不做考虑，达到最小频繁度阈值的事件集称为频繁事件集；最大事件窗口阈值避免了挖掘出来的故障序列模式中事件之间相隔过多，事件之间关联度很小，对预防性维修指导性不够的情况；最小支持度阈值指频繁事件集之间关联程度最低要求；最小共现度阈值避免了挖掘出来的故障序列模式只在小部分产品中频繁发生，不具有普遍性的情况。

a)事件。事件记做 e_i，满足，$e_i \in p$其中 p 为是事件集合。每个事件有一个事件时间标识，表示事件的顺序。

b)事件序列。序列记做$e_i \rightarrow e_j$，其中e_i、e_j表示事件 i 与 j。一个序列中的事件有时间先后关系，e_i 出现在 e_j 之前。

c)事件窗口。事件窗口记做win_{ij}，表示事件序列$e_i \rightarrow e_j$之间间隔的事件数目。

d)事件相似度。事件相似度记作X_{ij}，表示事件 i 与 j 之间相似的程度。

e)相似事件集。相似事件集记做$SES_k = [e_1^k, e_2^k, \ldots, e_n^k]$，其中$e_i^k$表示相似事件集$SES_k$中的 i 事件，集合中任意两个事件均满足最小相似度阈值 min_sim。相似事件集中的所有事件被认定为同一类事件。

f)相似事件集频繁度。事件频繁度记做$freq(k)$，是指相似事件集SES_k中事件的个数。

g)频繁事件集。当相似事件集SES_k中事件的个数$freq(k)$大于或等于min_freq时，相似事件集SES_k被认定为频繁事件集，记作$FES_k = [e_1^k, e_2^k, \ldots, e_n^k]$。其中 min_freq 表示最小频繁度阈值。

h)序列支持度。若存在事件序列$e_i^p \rightarrow e_j^q$，e_i^p、e_j^q分别表示 i 和 j 事件，分别属于频繁事件集FES_p、FES_q，且win_{ij}小于或等于最大事件窗口阈值max_win，则认定序列模式 $p \rightarrow q$的支持度加一，序列模式支持度记作$sup(p \rightarrow q)$，具体计算过程如后所示。

i)序列的共现度。一个序列$p \rightarrow q$的共现度是指该序列出现在不同产品的个数，序列 $p \rightarrow q$共现度可以表示为$occ(p \rightarrow q)$。

j)序列模式（sequential pattern）。序列$p \rightarrow q$是一个序列模式，当且仅当满足以下三个约束条件：a)p、q都是满足 min_freq 的 FES；b)$sup(p \rightarrow q) \geq min_sup$；c)$occ(p \rightarrow q) \geq min_occ$。其中 min_sup 指最小支持度阈值，min_occ 指最小共现度阈值

1.2 文本相似度衡量模型

由于自然语言的特性，不同个体对同一件事情的描述可能有所差别，完全相同的文本信息很少，进而难以找出故障序列模式。先对故障文本描述进行相似度的衡量，以更好的完成序列模式的挖掘。

1.2.1 文本预处理

使用语言模型进行自然语言处理是建立在词的基础上的而对于中文，词之间没有明确的分界符。因此需要先对故障文本进行分词，才能做进一步的自然语言处理。分词结果中存在一些区分度不高的介词，连词，标点符号等，为了更好地衡量文本相似度，需要进行去停用词的处理，经过分词的结果进行去停用词的处理。

1.2.2 文本向量化

词的分布式表示是指一个稠密的低维的实数向量。例如[0.792, −0.177, −0.107, 0.109, −0.542, …]。利用 Doc2Vec 模型将每一个分完词的句子被映射成一个独立的向量，Doc2Vec 模型能表示词和词之间的语义关系 ，考虑了词的先后顺序，能够很好地将文本向量化。本文使用的 Doc2Vec 训练模型如下：

model = Doc2Vec(sentences, size, window, min_count, workers, min_alpha)

其中 sentences：句子库；size：特征向量维度；window：要预测的词和文档中用来预测的上下文词之间的最大距离；alpha：初始学习速率；min_count：忽略总频数小于此的所有的词；workers：用来训练模型的电脑线程数量。

1.2.3 相似度计算

对于已经向量化的文本，我们利用余弦相似度进行文本之间相似度计算，余弦相似度通过测量两个向量的夹角的余弦值来度量它们之间的相似性。计算公式如下：

$$similarity = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (1)$$

其中：A_i 和 B_i 分别代表向量 A 和 B 的各分量。由相似度衡量的特性，可知相似度矩阵是对角线上均为 1 的对称矩阵。

1.3 算法流程

本算法设计流程如下主要包括建立故障文本相似度衡量模型和设计故障序列挖掘算法两部分。在故障文本相似度衡量之前，首先要对故障文本信息进行相关处理。在此基础上进行故障序列模式挖掘，最后通过实例验证该算法算法。设计流程如图 2 所示。

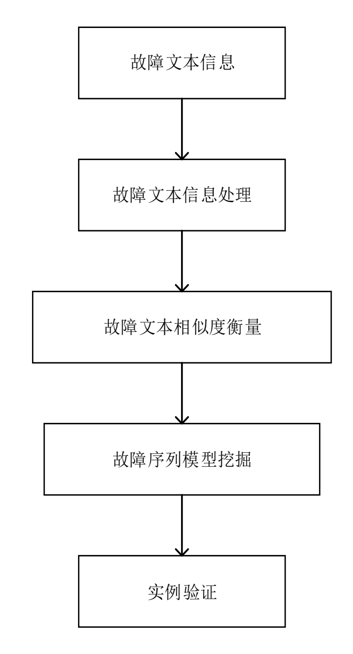


图 2 本文算法设计流程

2 算法构建

算法主要包括两部分，一是在文本相似度衡量基础上进行频繁事件集的挖掘，二是在频繁事件集基础上进行故障序列模式挖掘。

2.1 频繁事件集挖掘

2.1.1 算法描述

a)通过故障文本相似度衡量模型得到如表 1 所示的相似度矩阵。

表 1 文本相似度矩阵

事件	1	2	…	n
1	1	X_{12}	…	X_{1n}
2	X_{21}	1	…	X_{2n}
…			…	
n	X_{n1}	X_{n2}	…	1

其中X_{ij}表示事件 i 与事件 j 的相似度，显然当 i=j 时X_{ij}=1。

b)找出相似事件集，给定最小相似度阈值 min_sim 及式(2)。

$$\begin{cases} X_{ij} \geq min_sim,\ X_{ij} = 1 \\ X_{ij} < min_sim,\ X_{ij} = 0 \end{cases} \quad (2)$$

通过式（2）转换得到如表 2 所示的相似事件集矩阵。

表 2 转化后的相似度矩阵

事件	1	2	…	n
1	1	0 或 1	…	0 或 1
2	0 或 1	1	…	0 或 1
…	…	…	…	…
n	0 或 1	0 或 1	…	1

显然矩阵中值为 1 代表两事件为相似事件，属于同一事件集合，值为 0 代表两事件不相似，从而找出相似事件集SES_k。

c)找出频繁项目集，下面计算相似事件集频繁度：

$$freq_p = \sum_{j=1}^{n} X_{ij}\ ,\ i, p = 1,2, \dots, n \quad (3)$$

其中X_{ij}表示时间 i 与事件 j 的相似度，X_{ij}=1 或 0，$freq_p$表示第 i 个事件所在相似事件集SES_p的事件个数。

给定最小频繁度阈值 min_freq，

$$\begin{cases} freq_p \geq min_freq,\ \text{保留事件集}p \\ freq_p < min_freq,\ \text{去除事件集}p \end{cases} \quad (4)$$

根据式(4)得到若干频繁事件集FES_k。

2.1.2 算法流程

频繁事件集挖掘算法，流程如图 3 所示。

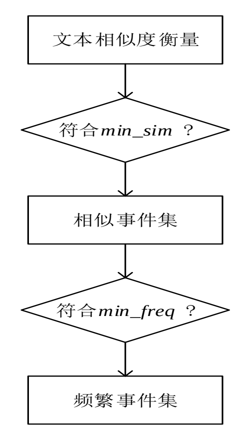


图 3 频繁事件集挖掘流程

2.2 序列模式挖掘

2.2.1 算法描述

a)对所有频繁事件集FES_k中的事件按照飞机 ID 进行划分，划分结果如表 3 所示。

表 3 频繁事件集

FES	ID		
	1	…	n
1	e_{1a}^{1}	…	e_{nr}^{1}
2	e_{1b}^{2}	…	e_{ns}^{2}
…	…	…	…
m	e_{1e}^{m}	…	e_{nt}^{m}

其中e_{ij}^{p}表示发生在第 i 架飞机上的第 j 个故障事件，并且故障事件 j 属于频繁事件集 p。

b)对单架飞机进行故障序列模式挖掘，方法如下：

在第 i 架飞机中，对于频繁事件集 p 与频繁事件集 q 中的故障序列，p 中的故障事件为e_{ia}^{p}，q 中的故障事件为e_{ib}^{q}，给定最大窗口事件 max_win，

$$if: \ win_{a\to b} \le max_win, \ \begin{cases} \sup(p\to q)=\sup(p\to q)+1 \\ occ(p\to q)=occ(p\to q)+1 \end{cases} \quad (5)$$

其中$win_{a\to b}$表示故障 a、b 之间间隔的故障事件数，$\sup(p\to q)$表示序列模式$p\to q$的支持度，$occ(p\to q)$表示序列模式$p\to q$的共现度。

利用该方法，算出序列模式$p\to q$在该架飞机上的的支持度和共现度。依次迭代，计算出所有频繁事件集之间序列模式的支持度和共现度。

c)迭代步骤 b)，依次对每架飞机进行故障序列挖掘。将序列模式在不同飞机上的支持度、共现度累加。

d)检验各序列模式是否满足最小支持度阈值、最小共现度阈值。对于序列模式$p\to q$，若满足式(6)

$$\begin{cases} sup(p\to q)\ge min_sup \\ occ(p\to q)\ge min_occ \end{cases} \quad (6)$$

其中 min_sup 表示最小支持度阈值，min_occ 表示最小共现度阈值，则认定序列模式$p\to q$成立。

2.2.2 算法流程

序列模式挖掘算法流程如图 4 所示。

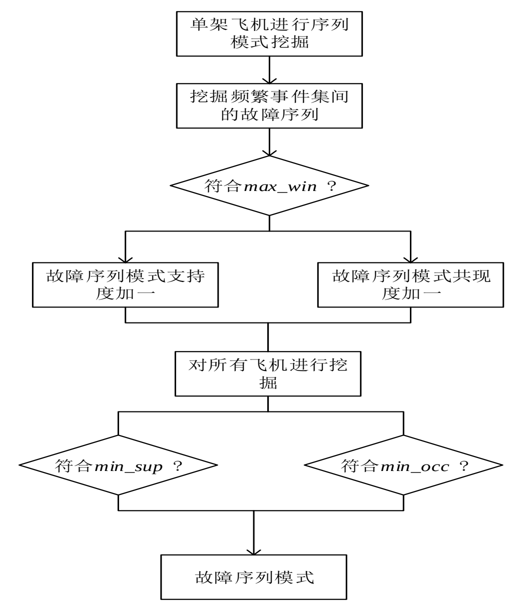


图 4 序列模式挖掘流程

3 实例验证

使用某型 3 架飞机共计 20 条故障情况文本信息作为实例对象，其中第 1 架飞机和第 2 架飞机分别有 7 条故障情况文本描述，第 3 台产品有 6 条故障情况文本描述，需要在其中找出故障序列模式。对应产品 ID 和故障序号在表 4 中列出。

表 4 故障文本描述

FES \ ID	1	2	3
1	2 发滑油散热器蜂窝孔渗油	4 发滑油散热器蜂窝结构渗油	3 发滑油散热器蜂窝结构漏油
2	4 发滑油散热器蜂窝结构渗油	起动发电机起动电流超差	2 发滑油散热风门不工作
3	3 发滑油散热器风门电机有卡滞现象	3 发滑油散热器蜂窝结构漏油	4 发滑油散热器蜂窝结构漏油
4	无线电高度表照明灯不亮	2 发滑油散热器蜂窝孔渗油	4 发航前起动电流差大
5	2 发滑油散热器风门不工作	无线电高度表指示器照明灯不亮	2 发滑油散热器蜂窝孔渗油
6	右 1 组油量表传感器多指 1000KG	2 发滑油散热风门不工作	3 发滑油散热风门不能自动关闭
7	3 发滑油散热器蜂窝结构漏油	3 发散热器蜂窝结构漏油	

3.1 文本预处理

故障文本示例如表 5 所示。

表 5 故障文本示例

故障情况文本（共计 20 条）
2 发滑油散热器蜂窝孔渗油
4 发滑油散热器蜂窝结构渗油
3 发滑油散热器风门电机有卡滞现象
无线电高度表照明灯不亮

对故障文本进行分词、去停用词，利用 Python 语言中的 jieba 分词包进行处理，得到结果如表 6 所示。

表 6 预处理之后的文本

故障情况文本
2 // 发滑油 // 散热器 // 蜂窝 // 孔 // 渗油
4 // 发滑油 // 散热器 // 蜂窝 // 结构 // 渗油
3 // 发滑油 // 散热器 // 风门 // 电机 // 有卡滞 // 现象
无线电 // 高度表 // 照明 // 灯不亮

3.2 计算故障文本相似度

运用 Doc2Vec 模型进行文本表示，其中模型参数选择如下：*model=Doc2Vec(sentences,size=10,window=3,min_count=3, workers=4, min_alpha=0.002)*

使用余弦相似度来进行相似度的衡量，结果如下：

$$\begin{bmatrix} 1 & 0.391 & 0.025 & \cdots & 0 & 1 & 0.008 \\ 0.391 & 1 & 0.028 & \cdots & 0.182 & 0.391 & 0.009 \\ 0.025 & 0.028 & 1 & \cdots & 0 & 0.025 & 0.108 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0.182 & 0 & \cdots & 1 & 0 & 0 \\ 1 & 0.391 & 0.025 & \cdots & 0 & 1 & 0.008 \\ 0.008 & 0.009 & 0.108 & \cdots & 0 & 0.008 & 1 \end{bmatrix}$$

3.3 故障频繁项目集挖掘

设定最小相似度阈值 $min_sim = 0.8$，将故障文本描述的相似度矩阵转化为 0-1 矩阵来方便频繁度的计算。维度为 20×20 的 0-1 矩阵如下：

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

各条故障文本描述的频繁度为：[3, 3, 1, 2, 3, 1, 4, 3, 1, 4, 3, 2, 3, 1, 4, 3, 6, 1, 3, 1]。

设定最小频繁度阈值 $min_freq = 3$，则利用伪代码程序计算可以得到频繁事件集文本序号为[1, 2, 5, 7, 8, 10, 11, 13, 15, 16, 17, 19]。

相似频繁项目集结果如表 7 所示。

表 7 频繁事件集

序号	飞机 ID		
	1	2	3
1	1	11	19
2	2	8	17
5	5	13	16
7	7	10	[15, 17]
8	2	8	17
10	7	10	[15, 17]
11	1	11	19
13	5	13	16
15	7	10	[15, 17]
16	5	13	16
17	[2, 7]	[8, 10]	[15, 17]
19	1	11	19

3.4 序列模式挖掘

设定最大事件窗口阈值 max_win=4，最小支持度阈值 min_sup=4，产品最小共现度阈值 min_occ=2，利用伪代码程序计算可以得到挖掘出故障序列模式，结果如表 8 所示。

表 8 故障序列模式挖掘结果

	序列模式	前向故障	后向故障	sup	occ
1	17→1	4 发滑油散热器蜂窝结构漏油	2 发滑油散热器蜂窝孔渗油	4	2
2	17→11	4 发滑油散热器蜂窝结构漏油	2 发滑油散热器蜂窝孔渗油	4	2
3	17→19	4 发滑油散热器蜂窝结构漏油	2 发滑油散热器蜂窝孔渗油	4	2

可以得到，满足条件的序列模式为{“4 发滑油散热器蜂窝结构漏油”⟶“2 发滑油散热器蜂窝孔渗油”}，根据序列模式支持度和共现度的结果来看，在该算法框架下该序列模式共发生 4 次，在 2 架飞机上出现过。通过与文本数据验证，找出的该故障序列关系是客观存的。根据结果来看，在产品的维修保障过程中，如果有发动机的滑油散热器发生漏油或渗油等问题，各个发动机的滑油散热器都应该去做检查，做到防患于未然。

4 结束语

文本记录的故障信息，其表达形式各异，结构化程度差，现有的序列模式挖掘算法不能直接对文本数据进行挖掘。针对文本信息，本文提出了一种基于文本相似度衡量的故障序列模式挖掘算法，其与现有方法的区别如下：

a) 由于文本信息的特殊性，相同信息的表达千差万别，本文提出的文本相似度衡量模型，可以有效的将相同意义的故障文本归为一类，将结构化程度差的文本数据整合归类，在此基础上进行序列模式挖掘。

b)本文针对文本数据提出了最小频繁度概念，针对序列模式挖掘提出了最大事件窗口、最小共现度两个概念，可以确保挖掘出的序列模式是普遍存在的。

实例验证表明，本文提出的算法是正确有效的，对于识别故障时间关系，进行故障预测和维修决策提供支持。

参考文献：

[1] 吴熠潇. 中文分词相关算法研究 [J]. 科技经济导刊, 2018 (2): 122-123.

(Wu Yixiao. Research on Chinese word segmentation algorithm [J]. Technology and Economic Guide, 2018 (2): 122-123.)

[2] 郑文超，徐鹏．利用 word2vec 对中文词进行聚类的研究 [J]. 软件, 2013, 34 (12): 160-162. (Zheng Wenchao, Xu Peng. Research on Chinese word Clustering with word2vec [J]. Software, 2013, 34 (12): 160-162.)

[3] 张志昌，周慧霞，姚东任，等．基于词向量的中文词汇蕴涵关系识别 [J]. 计算机工程, 2016, 42 (2): 169-174. (Zhang Zhichang, Zhou Huixia, Yao Rendong, *et al.* Recognition of Chinese lexical entailment relation based on word vector [J]. Computer Engineering, 2016, 42 (2): 169-174.)

[4] 周练. Word2vec 的工作原理及应用探究 [J]. 图书情报导刊, 2015, 25 (2): 145-148. (Zhou Lian. Exploration of the working principle and application of word2vec [J]. Sci-tech Information Development and Economy, 2015, 25 (2): 145-148.)

[5] 唐明，朱磊，邹显春．基于 Word2Vec 的一种文档向量表示 [J]. 计算机科学, 2016, 43 (6): 214-217. (Tang Ming, Zhu Lei, Zou Xianchun. Document vector representation based on word2vec [J]. Computer Science, 2016, 43 (6): 214-217.)

[6] Zhang Dongwen, Xu Hua, Su Zencai, *et al.* Chinese comments sentiment classification based on word2vec and SVM perf [J]. Expert Systems with Applications, 2015, 42 (4): 1857-1863.

[7] 殷耀明，张东站．基于关系向量模型的句子相似度计算 [J]. 计算机工程与应用, 2014, 50 (2): 198-203. (Yin Yaoming, Zhang Dongzhan. Sentence similarity computing based on relation vector model [J]. Computer Engineering and Applications, 2014, 50 (2): 198-203.)

[8] Tian Junfeng, Lan Man, Wu Yuanbin, *et al.* An adversarial joint learning model for low-resource language semantic textual similarity [C]// Advances in Information Retrieval. 2018: 89-101.

[9] 苗雪连．间隙约束序列模式挖掘的对比研究 [J]. 网络安全技术与应用, 2017 (2): 66-67. (Miao Xuelian. Comparative study of sequential pattern mining with gap constraints [J]. Network Security Technology and Application, 2017 (2): 66-67.)

[10] Krishna B. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Pattern [J]. International Journal of Computer Science & Engineering Survey, 2012, 2 (4): 111-122.

[11] Le B, Hai D, Truong T, *et al.* FCloSM, FGenSM: two efficient algorithms for mining frequent closed and generator sequences using the local pruning strategy [J]. Knowledge & Information Systems, 2017, 55 (3): 1-37.

[12] Yan Xiaowu, Zhang Jifu, Xun Yaling, *et al.* A parallel algorithm for mining constrained frequent patterns using MapReduce [J]. Soft Computing, 2017, 21 (9): 2237-2249.

[13] 武优西，周坤，刘靖宇，等．周期性一般间隙约束的序列模式挖掘 [J]. 计算机学报, 2017, 40 (6): 1338-1352. (Wu Youxi, Zhou Kun, Liu Jingyu, *et al.* Mining Sequential Pattern with Periodic General Gap Constraints [J]. Chinese Journal of Computers, 2017, 40 (6): 1338-1352.)

[14] Mooney C, Roddick J. Sequential pattern mining: approaches and algorithms [J]. ACM Computing Surveys, 2013, 45 (2): 1-39.

[15] Aloysius G, Binu D. An approach to products placement in supermarkets using PrefixSpan algorithm [J]. Journal of King Saud University of Computer & Information Sciences, 2013, 25 (1): 77-87.

[16] Wright A, Wright T, Mccoy A, *et al.* The use of sequential pattern mining to predict next prescribed medications [J]. Biomedical Informatics, 2015, 53 (C): 73.